Machine Learning Techniques for Music Prediction

S. Grant Lowe

March 19, 2015

Abstract

The question of whether one can predict the success of a musical composition has been asked many times over the years [14]. Researchers looking into hit song science, as it is called, have been able to predict whether a song will chart with varying degrees of success. However, not many have tried to look at how successful pop songs have changed over time and whether these changes can be tracked, to see if we are able to predict what popular songs in the future will be like. In my project, I will be looking at some of the attributes of popular music from the early 1900s to today, and seeing how those attributes change or become more important in determining a song's release year and genre. Using the popular data mining and machine learning framework WEKA[1], I hope to be able to track what attributes of a song give insight into the genre that that song falls into and the decade in which the song was released.

Contents

List of Figures

List of Tables

1 Introduction

Music has changed much over the course of human history. From social drum circles in ancient times to modern electronic music, humans have come up with many different ways to make sounds fit together. Once songs were able to be recorded and replayed over and over again, music became an even bigger part of human life, allowing us to listen whenever we pleased and make the music business a lucrative one. Now, songs can be recorded in released in a matter of days, and can become chart-toppers overnight. This leads to more songs, more genres, and more popular music than ever before.

Hit Song Science tries to identify what exactly makes a song popular [14]. Researchers in the past have tried numerous methods in order to figure out what makes a song a success [4], [5], [10]. Predicting and identifying features of popular music can tell us a lot about the music industry, human psychology and the culture of the area in which the music was produced. Much of the research in the field of Hit Song Science has been able to somewhat accurately predict whether a song would chart (usually within a range of positions) in today's musical climate.

Additionally, much work has gone into the fields of year[7] and genre prediction[18]. Studying the attributes that give insight into the year that song was released is important; it can show the evolution of music on a more precise level than ever before. Genre classification, while classically a very hard problem, gives insight into both the psychological aspect of analyzing music and the way in which that can be modeled. Genres of music change, evolve, and come into being at different points of time, which could be helpful in trying to determine when a song came out (for instance, if one was trying to find the year in which a song was released, and that song can be determined to be a 'grunge' song, then you might be able to also say that that song is from the 90's, as that is when that genre was the most popular).

Genre classification is historically a difficult problem for a multitude of reasons. Firstly, humans are not great at it, due to the fact that genre labels are essentially arbitrary. Humans think two songs sound similar stylistically and try to classify them as such. But many genres overlap; for instance, the difference between Alternative Rock, Grunge, and Rock is very small. Additionally, sometimes the same genre will develop independently in different locations and music scenes. For instance, punk rock from California is incredibly different from punk rock from London.

Year prediction has also been attempted by numerous researchers. These researches attempt to find out what makes the music from each year sound the way it does; they look at what attributes change over time and try to apply models that show this change [7]. Tracking these changes has numerous applications in both the writing and sales of music, as it allows composers and distributors to be able to quantitatively determine whether their song has a chance at being relevant when it is released.

This exploratory research has many real-world applications and can be useful for many fields. Being able to predict musical trends can help businesses decide what kind of artists to look for, along with giving artists an opportunity to really quantify how society's musical preferences change over time, which can aid in the songwriting process. Additionally, it could help for music services that attempt to find new music for people to listen to based on their preferences by finding new ways to classify the genres of songs.

2 Background and Related Work

There has been a lot of research in the musicology and computer science fields that tries to identify and solve similar problems. Researchers have been able to use data mining and machine learning for such topics as music genre classification, musical prediction, and hit song science. In a paper called Hit song Science Once Again a Science?, [14], researchers describe a method in which they determined which features of a song make it a chart success, and found out many interesting things about popular songs, such as the fact that songs have gotten longer on average as time has passed. Cherla et al. [15] was able to identify a process in which, given a sample of music, could predict melodic sequences of the rest of the song, with some degree of success. These sorts of processes give insight into how songs are composed, and can help researchers determine many facts about a song, without necessarily having all of the data available to them.

Other music classification research has found that both loudness and length of songs have been increasing over time. It is easy to see why these attributes would be important; music in general has gotten 'heavier', or less mellow and more energetic, so an increase in average decibel level is expected. Additionally, the average length of musical compositions is always changing; the songs heard on radio are, in general, longer, albeit less musically diverse (most radio-friendly songs have been sticking to the verse-chorus formula, and more progressive compositions are getting less popular). By exploring all these attributes further, I will be able to build much better models.

3 Methods

I used WEKA [1] and the Million Song Dataset (MSD) [7] to conduct the majority of my research. Additional work was done using python and related modules, in order to compile a version of the MSD for usage in WEKA.

3.1 WEKA

WEKA (which stands for Waikato Environment for Knowledge Analysis) is a suite of machine learning software. It was developed at the University of Waikato in New Zealand, and is used for Union's Data Mining and Machine Learning course.

WEKA contains visualization tools and algorithms for data analysis and predictive modeling. It was written in Java, and allows the user to do many standard data mining tasks, such as data preprocessing, clustering, classification, regression, visualization, and feature selection. The user is able to simply import a file in a supported format, and is then able to use a GUI in order to accomplish these tasks[1].

WEKA allows the user to pre-process data; it gives filters that can transform the data (for tasks such as discretization), and remove or ignore attributes that are not useful for the chosen classification or regression algorithms. WEKA can apply many algorithms to any given dataset, and is then able to estimate the accuracy of the model. WEKA also allows the user to attempt to identify relationships between attributes in the data, using association rule learners, and cluster the data[1]. The first figure below shows what WEKA looks like after applying filters to an attribute. Figure 2 shows an example of WEKA output.

3.2 Million Song DataSet

In order to do my research, I looked for a comprehensive dataset that had many different auditory and informational features for its music data. What I eventually settled on is the Million Song Dataset[7], a freely-available collection of audio features and metadata for a million contemporary popular music tracks.

Selected attribute

Name: year

Missing: 4197 (53%)

Distinct: 67

Type: Nominal

Unique: 8 (0%)

No.	Label	Count
1	1927	3
2	1929	1
3	1930	2
4	1934	1
5	1935	1
6	1936	1
7	1940	2
8	1947	2
9	1950	1
10	1953	4
11	1954	1
12	1955	1
13	1956	5
14	1957	1
15	1958	4
16	1959	5
17	1960	9
18	1961	6
19	1962	4
20	1963	12
21	1964	10

Figure 1: A picture of the Million Song Dataset's Year Distribution, after being nominalized by a WEKA filter

The Million Song Dataset was developed by Columbia University's LabROSA and The Echo Nest, which is a music intelligvence company that aims to provide "developers with the deepest understanding of music content and music fans". The Echo Nest provides services for companies such as Spotify, SiriusXM, VEVO, and Microsoft, among many others. The Million Song Dataset was created under a grant from the National Science Foundation.

3.3 Attributes

The songs featured in the data set contain 54 attributes. These attributes will be described in detail later on. The full dataset is about 273 GB of data, with over 44,000 unique artists, and 515,000 dated tracks, starting from 1922. The dataset is distributed in the HDF5 data format, which is a data model, library, and file format

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         370               9.9462 %
Incorrectly Classified Instances      3350              90.0538 %
Kappa statistic                          0.0606
Mean absolute error                      0.0269
Root mean squared error                  0.164
Relative absolute error                 93.6509 %
Root relative squared error            136.8904 %
Total Number of Instances             3720
Ignored Class Unknown Instances                4197
```

Figure 2: Sample WEKA output.

for storing and managing data. The HDF Group describes the format (on their website, hdfgroup.org) thusly: "It supports an unlimited variety of datatypes, and is designed for flexible and efficient I/O and for high volume and complex data. HDF5 is portable and is extensible, allowing applications to evolve in their use of HDF5. The HDF5 Technology suite includes tools and applications for managing, manipulating, viewing, and analyzing data in the HDF5 format." [6]

4 Design

4.1 Preliminary Work

The first step in this work was to download a subset containing approximately 10,000 songs from the MSD website. This smaller set was used during the early phase of work due to the fact that it made it easier to manage and find preliminary trends trends., Additionally, this subset was only 1.8GB, as compared to the full sets 280GB. A script was written, using python and some python modules, notably pytables and pandas, in order to convert the HDF5 file into a WEKA-friendly CSV. Pandas [8] is a module that allows the user or program to read just the pertinent information they want from a large dataset in HDF5 format, using a dict-like HDFStore class. This module contained support for exporting to CSV. WEKA also has problems reading files with strange ASCII characters, so the converted CSV had to be edited by hand in order to make it WEKA-friendly. Early tests were run using the resulting CSV file, which contained about 4,000 songs; the other 6,000 songs were missing data (mostly the year attribute, which was what was being tested for), so they were discarded by the python script.

4.2 Hypothesis

During this period, the hypothesis for my research was formed:

- Given a song, can a machine learning algorithm predict the year that song was released?
  - Given a song's BPM, key, duration, segment duration, time signature, and mode can the decade in which a song was released be determined?
  - Given a song's pitches, "segment start" value [said to be an array of musical events/notes], and segment's timbre, can the year in which a song was released be determined?
  - If not the year, what about the decade?
  - How are musical decades defined?
- Given a song, can the genre of that song be predicted?
  - What are the correct genre labels for music?
  - How is genre prediction similar to year prediction?
- Can we identify which attributes, if any, are the strongest in answering these questions?

4.3 Early Challenges

There were many challenges in this first term of real research. The main problem was that the subset I was working with was so small (only around 4,000 songs). This means that my data could possibly not be representative of the full set. Additionally, the distribution of songs from each year was heavily skewed, with the songs from before 1950 having very little representation, while most songs in the 80s and 90s had hundreds of instances (in the full set, thousands). Figure 2 shows this skewed distribution. Additionally, many of the attributes had to be ignored due to being missing for many of the songs.

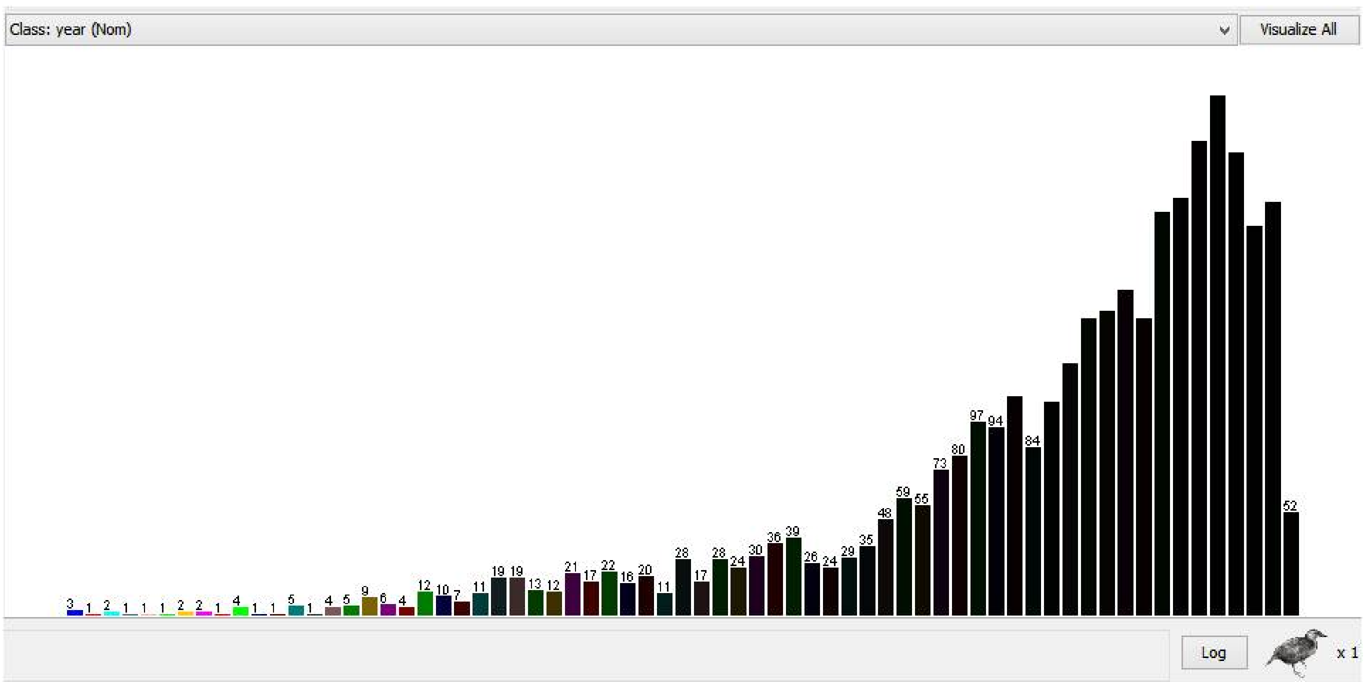
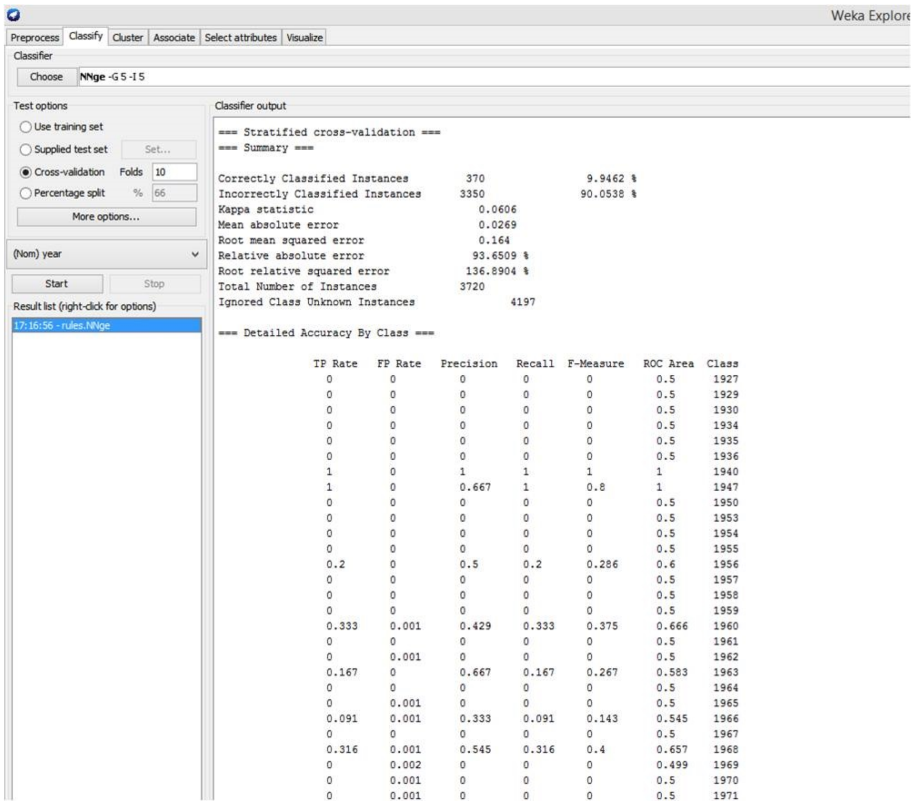


Figure 3: A picture showing the distribution of the release years of the songs in the sub-set

4.4 Early Results

At first, I tried to run WEKA tests with all 54 attributes. However, I found that many of these attributes were unnecessary in trying to determine a song's year. I also noticed that there were attributes that were empty for every song. So, after careful thought, I was left with 19 fields, listed with their descriptions below. These 19 fields were further narrowed down based on preliminary testing. The method of discretization of attributes was used in a few ways. Discretization is the process of turning numeric attributes into nominal ones; in other words, turning a numeric attribute like 1979 into a class, the 1970s. Tests were run discretizing



TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0	0	0	0	0	0.5	1927
0	0	0	0	0	0.5	1929
0	0	0	0	0	0.5	1930
0	0	0	0	0	0.5	1934
0	0	0	0	0	0.5	1935
0	0	0	0	0	0.5	1936
1	0	1	1	1	1	1940
1	0	0.667	1	0.8	1	1947
0	0	0	0	0	0.5	1950
0	0	0	0	0	0.5	1953
0	0	0	0	0	0.5	1954
0	0	0	0	0	0.5	1955
0.2	0	0.5	0.2	0.286	0.6	1956
0	0	0	0	0	0.5	1957
0	0	0	0	0	0.5	1958
0	0	0	0	0	0.5	1959
0.333	0.001	0.429	0.333	0.375	0.666	1960
0	0	0	0	0	0.5	1961
0	0.001	0	0	0	0.5	1962
0.167	0	0.667	0.167	0.267	0.583	1963
0	0	0	0	0	0.5	1964
0	0.001	0	0	0	0.5	1965
0.091	0.001	0.333	0.091	0.143	0.545	1966
0	0	0	0	0	0.5	1967
0.316	0.001	0.545	0.316	0.4	0.657	1968
0	0.002	0	0	0	0.499	1969
0	0.001	0	0	0	0.5	1970
0	0.001	0	0	0	0.5	1971

Figure 4: A picture of preliminary results.

the year attribute into both individual years, and decade bins. This was done in order to turn the numeric year attribute into a nominal one, so that the years could be represented as groups, instead of numbers. Many different binning methods were also used. First, I was defining decades in the way that they are classically defined: when people are talking about the 1960s, they mean the years between 1960 and 1969. However, work was done to see if musical decades really match up so well with actual decades. The intuition here was that perhaps in the early years of a particular decade, the music still is stylistically similar to the music from the previous decade. In other words, perhaps when humans think of music from the 60s as having a particular sound, they are really thinking about music from 1963-1973. So, after finding the baseline classification results for the normal way we think about decades, the same test was done by rebinning the decades into 1961-1971, 1971-1981, etc., and then 1962-1972, 1973-1983 (and so on), up to

1969-1979, 1979-1989, etc. However, as it turns out, the classifier performed the best was the classic binning of decade (1960-1970, etc.), while the others performed significantly worse. So, what this tells us is that the way we classify musical decades matches up well with the actual way we think about decades.

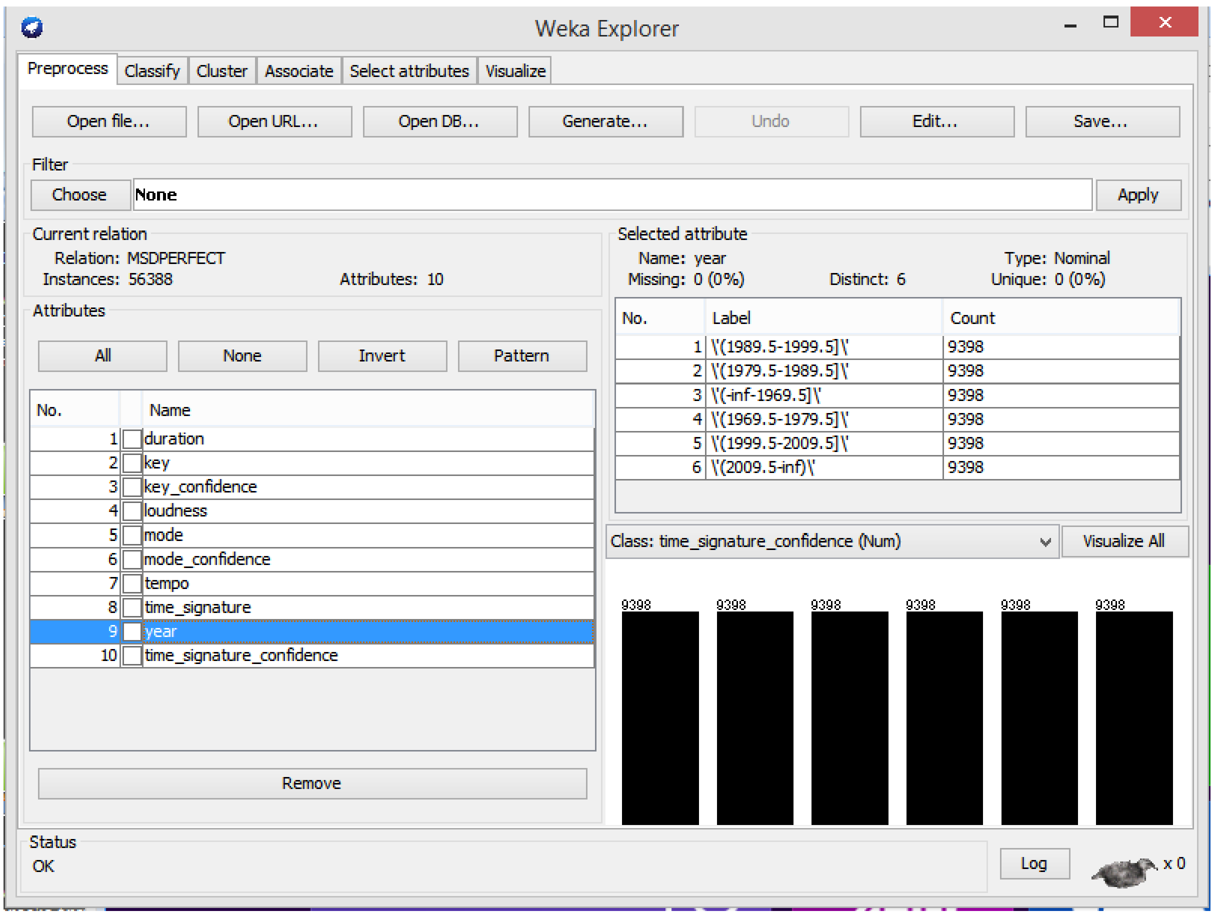


Figure 5: A picture of the dataset used for classification, with year binned into equal decades 1960-1970,...,2010-2020

4.5 Attribute Selection

Here is a list of the attributes that I looked at the most in-depthly and pruned from the original 54 attributes.

duration The duration of the song in seconds.

end of fade in Seconds at the beginning of the song (ie, how long you have to wait until the music starts).

key The key the song is in. The key identifies the tonic triad, the chord, major or minor, and represents the final point of rest in a piece.

key confidence confidence measure, key is algorithmically determined

loudness loudness of the song in dB

mode major/minor - 0 or 1.

mode confidence similar to key confidence

start of fade out when in the song the music stops

tempo estimated tempo in BPM

time signature number of beats per bar

time signature confidence similar to other confidence measures

artist familiarity algorithmically determined value for how familiar this artist is to the world (in 2010, when the dataset was created).

artist hottness similar to above. ”Hott-ness” refers to the popularity of a given artist when the dataset was created.

artist lat float, the latitude of the artists location

artist location artist’s location, string.

artist long float, the longitutde

artist name The name of the artist.

song hottnesss Similar to artist hotttnesss, but for song.

title Song title

year The year in which the song was released.

4.6 Timbre Attributes

The Million Song Dataset also featured 90 timbre attributes for each song. Timbre is defined as the quality of a musical note or sound that distinguishes different types of musical instruments or voices. It is a complex notion also referred to as a sounds color, texture, or tone quality, and is derived from the shape of a segments spectro-temporal surface, independently of pitch and loudness. Each song in the MSD was split into segments, from which a twelve dimensional feature vector was derived [23]. This vector contained information on each segments loudness, attack, decay, among other aspects of its timbre. The 90 Timbre attributes of the MSD represent the average and covariance across these feature vectors. MORE

4.7 Attribute Motivation

The main attributes that were first explored were are tempo, key (and mode, as they are related), and loudness. Early research has shown that tempo of popular songs has definitely increased. One of the first hypotheses that was explored was finding out if key and mode would be good attributes for year prediction. Key is represented numerically (0-11), representing the 12 major/minor keys. A second attribute, Mode, also represented numerically (0 or 1), tells whether it is the major or minor key. My reasoning here was to find out whether there is any evidence to say that the relative keys are popular in the same year, or if there is any Circle of Fifths styled pattern to what keys are popular in certain years/decades.

4.8 Removing Attributes

Attributes were further pruned from the data set in order to find the most important attributes in classifying for release decade and genre. Attributes were removed if they were too closely correlated to what was being classified for. An early example of this during the early stages was when it was noticed that the attribute ArtistID was too closely tied to decade; this is due to the fact that most artists are only active in popular music for a fairly short amount of time - especially for more modern music. Additionally, attributes were removed if they had no bearing on classification accuracy. This occurred for attributes such as the fade-in/fade-out metrics (which measured the amount of silence before and after the song starts/ends, along with other impertinent attributes.

4.9 Final Datasets

For classification tests, three different datasets were used. For year prediction, I used one dataset with only the Descriptive attributes (listed below), with 9398 instances of songs from each decade (60s-2010s; the decades from before then were too under represented in the MSD in order to accurately test.) The other year prediction set contained the same songs, but with the 90 timbre attributes instead. The final dataset was used for genre prediction; this set contained the same songs, with their genre tags, along with the descriptive attributes and the timbre attributes.

4.10 Ranked Attributes

The descriptive attributes were eventually narrowed down to the following six, ranked in order of importance. The figure below shows this ranking, along with each attributes weight.

1. Loudness
2. Duration
3. Tempo
4. Time Signature
5. Key
6. Mode

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 9 year):
        Information Gain Ranking Filter

Ranked attributes:
 0.13872   4 loudness
 0.0825    1 duration
 0.03863   7 tempo
 0.02367  10 time_signature_confidence
 0.01725   8 time_signature
 0.01044   2 key
 0.00621   5 mode
 0.00301   3 key_confidence
 0.00185   6 mode_confidence

Selected attributes: 4,1,7,10,8,2,5,3,6 : 9
```

Figure 6: WEKA output showing the weights of each attribute.

5 Classification Results

5.1 Decade Classification

The MSD was discretized into six decades for decade prediction: 1960-1970, 1970-1980, and so on. The baseline for classification (chance selection) is 16.67 percent. This is the classification accuracy if one were to just randomly assign every song to a decade. For the descriptive attributes, the best classifier was a Bayesian Network, with a classification accuracy of 31.87 percent. For the timbre attributes, the Random-Forest classifier was able to predict decade with an accuracy of 40.18 percent. The figure below shows the

Experimenter Module of WEKA, displaying the results of a statistical significance test on the Descriptive Attributes best classifiers (and the ZeroR classifier, for reference.) The table below shows the performance of different classifiers on the Descriptive Attribute set.

Test output

```
Tester:     weka.experiment.PairedCorrectedTTester
Analysing:  Percent_correct
Datasets:   1
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       3/17/15 11:19 PM


Dataset                   (1) rules.ZeroR '' | (2) trees.Rando (3) bayes.Bayes
------------------------------------------------------------------------------
'MSDPERFECT - experimente(100)   16.65(0.00) |   27.54(0.63) v   31.79(0.57) v
------------------------------------------------------------------------------
                                     (v/ /*) |         (1/0/0)         (1/0/0)


Key:
(1) rules.ZeroR '' 48055541465867954
(2) trees.RandomForest '-I 10 -K 0 -S 1' 4216839470751428698
(3) bayes.BayesNet '-D -Q bayes.net.search.local.K2 -- -P 1 -S BAYES -E bayes.net.estimate.SimpleEstimator -- -A 0.5' 746037443258775954
```

Figure 7: The results of Statistical Significance Testing.

Table 1: Table showing the classification percentage of various classifiers.

Classifier	Percentage Correctly Classified
BayesNet	31.87
HyperPipes	16.67
IBK	22.00
libSVM	26.13
Multilayer Perceptron	29.11
NaiveBayes	27.55
RandomForest	27.54

5.2 Genre Classification

Genre was classified into the following ten discrete values [18]:

1. Classic Pop and Rock
2. Classical
3. Dance and Electronica
4. Folk
5. Hip-Hop
6. Jazz
7. Metal
8. Pop
9. Rock and Indie
10. Soul and Reggae

The baseline for genre classification is 10.00 percent accurately classified; this would occur if one were to just randomly assign each song a particular genre. The best classifier to be found was a Bayesian Network, with an accuracy of 41.55 percent.

```
Time taken to build model: 3.7 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       17971               31.8703 %
Incorrectly Classified Instances     38417               68.1297 %
Kappa statistic                          0.1824
Mean absolute error                      0.2561
Root mean squared error                  0.3597
Relative absolute error                 92.1849 %
Root relative squared error             96.5176 %
Total Number of Instances            56388

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.569     0.179      0.389     0.569     0.462      0.789    '(-inf-1969.5]'
                 0.335     0.16       0.296     0.335     0.314      0.681    '(1969.5-1979.5]'
                 0.206     0.099      0.295     0.206     0.243      0.633    '(1979.5-1989.5]'
                 0.178     0.113      0.24      0.178     0.204      0.594    '(1989.5-1999.5]'
                 0.155     0.075      0.291     0.155     0.202      0.697    '(1999.5-2009.5]'
                 0.469     0.192      0.328     0.469     0.386      0.724    '(2009.5-inf)'
Weighted Avg.    0.319     0.136      0.306     0.319     0.302      0.686

=== Confusion Matrix ===

    a    b    c    d    e    f   <-- classified as
 5347 1726  661  828  284  552 |    a = '(-inf-1969.5]'
 2371 3146 1421 1037  428  995 |    b = '(1969.5-1979.5]'
 2018 2127 1940 1251  620 1442 |    c = '(1979.5-1989.5]'
 1851 1589 1265 1672  879 2142 |    d = '(1989.5-1999.5]'
 1156 1074  670 1144 1454 3900 |    e = '(1999.5-2009.5]'
 1000  980  627 1049 1330 4412 |    f = '(2009.5-inf)'
```

Figure 8: WEKA output of best results.

5.3 Discussion of Classifiers

The two best classifiers for both Genre and Decade prediction were the RandomForest classifier and the Bayesian Network classifier. The RandomForest classifier works by constructing multiple decision trees during training and then outputting the class that is the mode of the classes of the trees. RandomForest helps correct for the fact that decision trees tend to overfit to the training set.

6 Conclusions and Future Work

Finding that loudness was the best predictor for decade prediction was interesting. Previous work had also noticed this trend, and it also ties into what is called The Loudness War. This is a term used to describe the trend that started first occurring as early as the 40s, but got more aggressive starting in the early 90s. With the advent of digital music encoding, producers would frequently compress their audio so that the decibel level of the recording can more frequently peak at the maximum amplitude allowed by the digital format [24].

The fact that the Timbre attributes were able to produce a more accurate classifier than the descriptive ones is an interesting observation which requires more research. Part of the reason for this could be that the increase in quality of recordings is affecting the values of the Timbre attributes, making that an important aspect over the songs composition.

Many music researchers have said we are living in a post-genre world due to the ever rapidly increasing rate at which new genres are born, popularized, and forgotten.[19] Because of this fact, it is incredibly important to study the ways in which music is put into categories. The question of what makes two songs similar in such a sense is one that begs to be answered, especially considering the ways in which the music industry could benefit from increased ability to categorize music.

7 References

1. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

2. Python Software Foundation. Python Language Reference, version 2.7. Available at http://www.python.org

3. P. J. P. de Leon, D. Rizo, R. Ramirez, and J. M. Inesta, Melody characterization by a genetic fuzzy system," ICML, 2008.

4. R. Nobrega and S. Cavaco, Detecting key features in popular music: Case study - voice detection," ECML PKDD, 2009.

5. G. Mitri, A. L. Uitdenbogerd, and V. Cielsielski, Automatic music classification problems," Australasian Computer Science Conference, 2004.

6. The HDF Group. Hierarchical Data Format, version 5, 1997-2014. http://www.hdfgroup.org/HDF5/.

7. Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011.

8. Wes McKinney. Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 51-56 (2010)

9. T. Perez-Garcia, C. Perez-Sancho, and J. M. Inesta, Harmonic and in- strumental information fusion for musical genre classification," ECML PKDD, 2010.

10. S. Dieleman and B. Schrauwen, Learning content-based metrics for music similarity," in 5th Internation Workshop on Machine Learning and Music, Abstracts, 2012.

11. D. J. Mason, A hierarchial cluster analysis to identify principles of popular music composition used by the beatles," ECML PKDD, 2013.

12. N. Andreas, P. Maria, R. Ioannou, N. Petkov, and C. N. Schizas, A machine learning approach for clustering western and non-western folk music using low-level and mid-level features," ECML PKDD, 2013.

13. J. J. Alaguraja, Musical data mining pattern matching apriori and dhp algorithm," in International Journal of Computer Applications, 2013.

14. Y. Ni, R. Santos-Rodriguez, M. McVicar, and T. D. Bie, Hit song science once again a science?" ECML PKDD, 2011.

15. S. Cherla, A. d'Avila Garcez, and T. Weyde, A neural probabilistic model for predicting melodic sequences," ECML PKDD, 2013.

16. Serra, J., Corral, A., Boguna, M., Haro, M., Arcos, J.L. (2012). Measuring the evolution of contemporary western popular music. Scientific Reports, 2. doi:10.1038/srep00521. Accessed 15 January 2015.

17. McFee and G.R.G. Lanckriet. Large-scale music similarity search with spatial trees. In Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR, pages 5560, 2011a.

18. Genre Paper from Week 2 Winter

19. Other Genre Paper from Week 2 Winter

20. The Echo Nest Taste profile subset, the official user data collection for the Million Song Dataset, available at: http://labrosa.ee.columbia.edu/millionsong/tasteprofile

21. Last.fm dataset, the official song tags and song similarity collection for the Million Song Dataset, available at: http://labrosa.ee.columbia.edu/millionsong/lastfm

22. http://blog.last.fm/2011/06/22/lyric-clouds-genre-maps-and-distinctive-words

23. http://developer.echonest.com/docs/v4/static/AnalyzeDocumentation.pdf

24. "The Loudness Wars: Why Music Sounds Worse". NPR. 31 December 2009

References